# IDG2003 – Lab 8

Exercise 1 : - In this lab, the task is to create a login system. Through the use of cookies and sessions, the user preferences and login details can be tracked.

(a) First, create the following database on PHPMyAdmin(simply execute the following MySQL commands on the platform) :

> CREATE DATABASE mysite;
>
> USE mysite;
>
> CREATE TABLE users(username VARCHAR(32), name VARCHAR(32), surname VARCHAR(32), gender VARCHAR(10), password VARCHAR(256), PRIMARY KEY(username));

(b) Create a script named Registration.php
1. In this script, create a form that ask the user, for a username, name, surname, gender and password.
2. Use HTTP POST to retrieve the data from the form.
3. Validate the user input, check for no empty fields.
4. Create a function to sanitise the user input to protect against both HTML and MySQL injection. (Recall the functions htmlentities() and mysqli_real_escape_string() we covered in lectures, but also consider using stripslashes() and strip_tags() functions)
5. When the user clicks on 'Register' button from the form, if the input is validated, a new user must be created on the database. Remember  : Its important to store the password as a hash token and not in plain text.

(c) Create another script named login.php containing another form which will allow an existing user to login.
1. Create a session before the user logs in.
2. Again, validate and sanitise the input from the user. (In this case, its only the username that would need sanitising).
3. Create a database function that would retrieve the stored password corresponding to the input username(if any), and crosscheck it against the password the user inputs.
4. If the password matches, Login the user and regenerate a new ID for the session for security reasons.
5. Set session parameters such as :
   i. username, name, surname, retrieved from the database
   ii. A Boolean 'isloggedin', which is set to TRUE if login was successful
   iii. IP address and the user agent of the user's machine(Refer to lectures).
6. In the html body of the login.php script, add a condition to check if the session parameter, isloggedin is set. If it is, log in the user automatically and provide a link to the profile.php script. If isloggedin is not set, display the login form.

(d) Create another script named profile.php
  1. Enable session
  2. If the 'isloggedin' parameter is not set OR the ip parameter does not match the Server's remote address OR the useragent parameter does not match the Server's Http user agent, force the page to go back to the login.php.
     Hint : The header() function would allow you to redirect to another page.
  3. Create a form which allows the user to select a color and store their choice in a cookie. When they refresh their profile page, the background's color should be the same as in the stored cookie. Note that you should set a default value for the color(such as #FFFFFF for white).
  4. Create another form which allows the user to log out.
     i. Here simply a logout button is required, where if pressed, it would empty the $_SESSION superglobal and destroy the session and redirect back to the login.php page.

To help you get started, I provided a functions file, containing the database connect and disconnect functions. A function to display the Login form on the login.php page and another function to validate the input are provided. You may add your own functions to that file.